Quelques éléments en matière de sécurité des cartes réseau

Loïc Duflot, **Yves-Alexis Perez**,
Guillaume Valadon, Olivier Levillain

Agence Nationale de la Sécurité des Systèmes d'Information

Les cartes réseau modernes...

...ne se contentent pas de connecter une machine à un réseau.

- Une architecture matérielle complexe :
  - plusieurs processeurs,
  - différents types de mémoires,
  - plusieurs interfaces réseau ;
- Les logiciels embarqués (*firmware*) comportent des fonctions :
  - d'administration à distance : `ASF`, `IPMI`, `AMT`, etc.
  - liées à la segmentation `TCP`,
  - de gestion du lien radio avec contraintes temps-réel : `GSM`, `802.11`, etc.

ANSSI

Conséquences sur la sécurité

Si un attaquant pouvait exécuter du code sur une carte, il pourrait pratiquement tout faire :

- arrêter le traitement des paquets ;
- ignorer certain paquets ;
- empoisonner les caches `ARP` et `DNS` ;
- implémenter des attaques comme *SSLstrip* ;
- rebondir vers d'autres machines du réseau ;
- remplacer le *firmware* ;
- attaquer l'hôte (via des accès en lecture/écriture à la mémoire centrale).

À lire sur ce sujet :
Arrigo Triulzi, PacSec08, « Project Maux Mk.II »

ANSSI

Ce dont on parlera aujourd'hui

- l'architecture des cartes réseau modernes ;
- un protocole d'administration à distance ;
- une vulnérabilité que nous avons mise en évidence ;
- des outils développés pour examiner les cartes *Broadcom NetXtreme* ;
- une preuve de concept et une démonstration ;
- comment se protéger contre ce type d'attaques.

ANSSI

Ce dont nous ne parlerons *pas* aujourd'hui

Cette étude ne traite pas :

- des failles dans les pilotes ;
- des vulnérabilités dans les systèmes d'exploitation.

À noter :

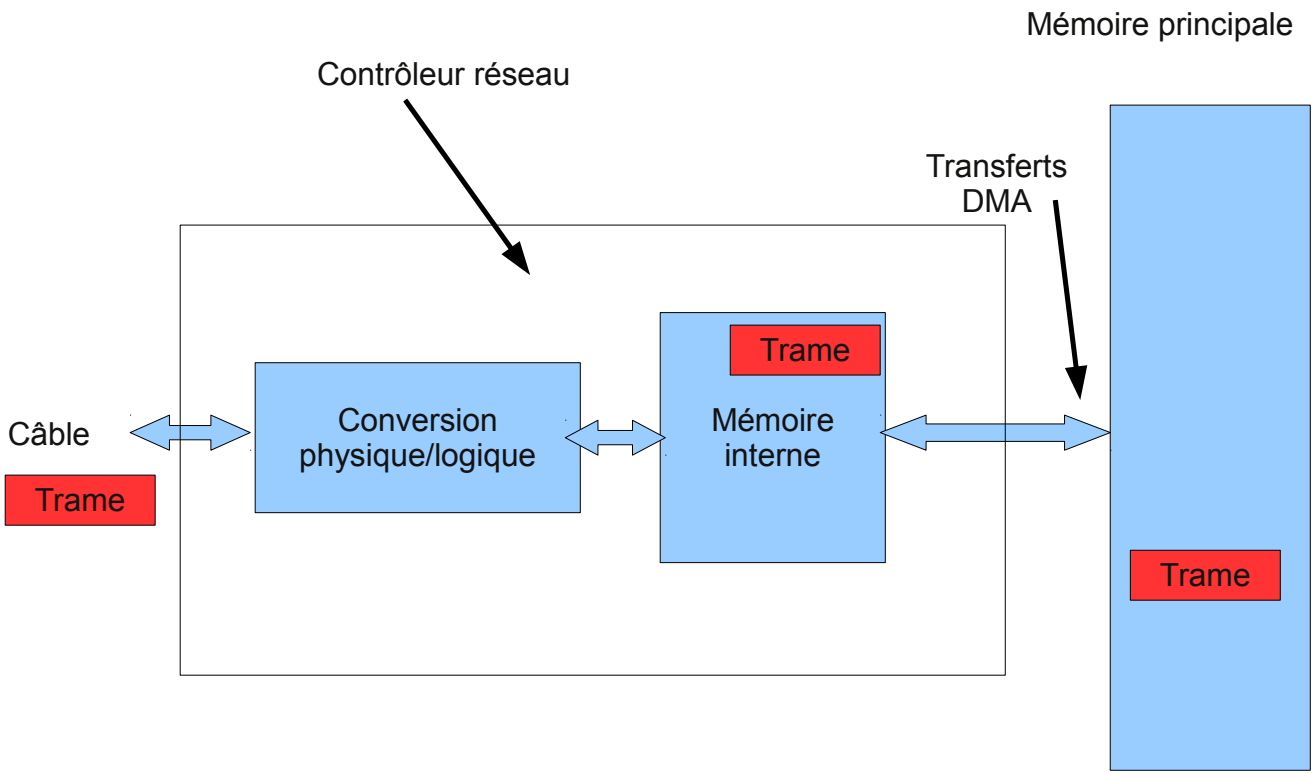
- nous ne fournirons pas les paquets et outils utilisés dans la démonstration ;
- nous avons travaillé avec les vendeurs et un correctif a été diffusé :
  - `CVE-2010-0104` : *HP Small Form Factor or Microtower PC with Broadcom Integrated NIC Firmware, Remote Execution of Arbitrary Code*

`http://www.ssi.gouv.fr/trustnetworkcard`

ANSSI

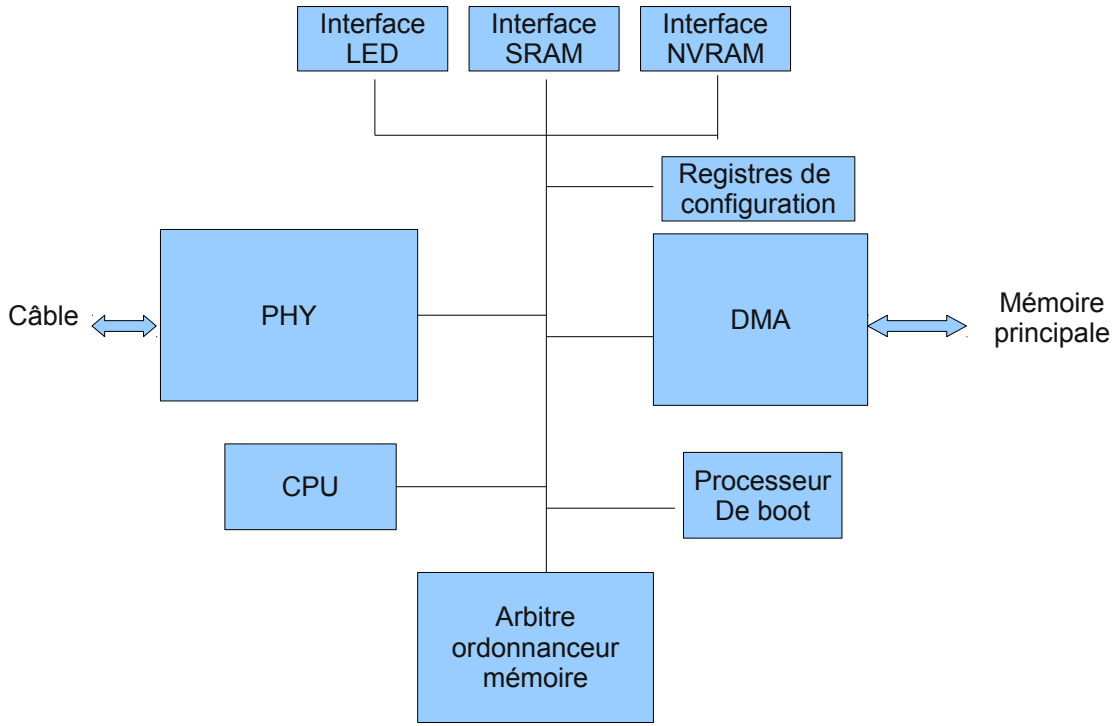
Architecture interne d'une carte réseau

- PHY : émission/réception du signal sur le cable ;
- moteur DMA : échange des paquets avec l'hôte ;
- négociation, contrôle du lien (vitesse, mode simple/duplex) etc.



Architecture interne d'une carte *NetXtreme*

- effectue diverses opérations sur les paquetspour décharger l'hôte de certaines tâches ;
- met en œuvre pour ce type d'opération des fonctions matérielles et de logicielles embarquées ;
- joue en fait le rôle de proxy transparent.



ANSSI

RX RISC

Extrait de la documentation : *An on-chip RISC processor is provided for running value-added firmware that can be used for **custom frame processing**. The on-chip RISC operates independently of all the architectural blocks ; essentially, RISC is available for the **auxiliary processing of data streams.***

- il y a un processeur `MIPS` sur la carte ;
- il peut accéder à tous les composants importants :
  - mémoire interne,
  - paquets entrants et sortants,
  - espace de configuration `PCI`,
  - `SMBus` (bus de configuration et d'administration) ;
- il exécute le code d'un *firmware*.

ANSSI

Les *firmwares* NetXtreme

Différents *firmwares* peuvent être chargés :

- `ASF` (*Alert Standard Format*) ;
- `TSO` (*TCP Segmentation Offloading*).

Le *firmware* est chargé :

- depuis l'`EEPROM` au démarrage de la carte ;
- ou depuis le système de fichiers par le pilote ;
  - seul le *firmware* `TSO` est disponible nativement dans le noyau Linux,
  - les pilotes Windows semblent n'avoir que rarement un *firmware*,
  - ceux-ci semblent être protégés ;
- vers la mémoire interne pour exécution.

ANSSI

Mémoire interne

- la mémoire interne de la carte est projetée dans celle de l'hôte, afin que le pilote puisse éventuellement y avoir accès ;
- l'accès se fait via une fenêtre de 32ko ;
- cette fenêtre peut parcourir toute la mémoire interne.

ANSSI

Alert Standard Format (ASF) 1.0 1/3

ASF

- transmet des événements et alertes via le réseau :
  - problèmes de disques, erreurs BIOS...
  - *heartbeats* ;
- doit fonctionner même si plus rien ne marche (disque dur hors service, système qui ne démarre pas...)

La carte reçoit les événements des autres périphériques via le SMBus (*System Management Bus*).

RMCP

ASF utilise un protocole appelé *Remote Management and Control Protocol* qui permet :

- d'interroger une machine sur l'état du système ;
- de démarrer, arrêter ou redémarrer une machine à distance.

ANSSI

Alert Standard Format (ASF) 1.0 (2/3)

- on configure certains paramètres dans le *firmware* : adresse IP et masque de sous réseau, fréquence du *heartbeat* ;
  - les vendeurs fournissent un utilitaire dédié ;
- ASF utilise une table ACPI spécifique ;
- sur certains systèmes, on peut désactiver ASF depuis le BIOS ;
- au moins un démarrage sur un système compatible ACPI est nécessaire.

ANSSI

ASF 1.0 (3/3)

Sécurité

- aucun mécanisme de sécurité n'est proposé dans les spécifications ;
- il est fortement déconseillé aux vendeurs implémentant ASF de développer des mesures de sécurité propriétaires ;
- les problématiques de sécurité doivent être prise en compte au niveau de l'infrastructure réseau (sic).

ANSSI

Alert Standard Format (ASF) 2.0

ASF 2.0 ajoute un nouveau protocole : RSP

- *RMCP security-extensions protocol* ;
- authentification et la protection des messages en intégrité ;
- pas de chiffrement.

Notre présentation concerne principalement ASF 2.0.

ANSSI

RMCP dans ASF 2.0

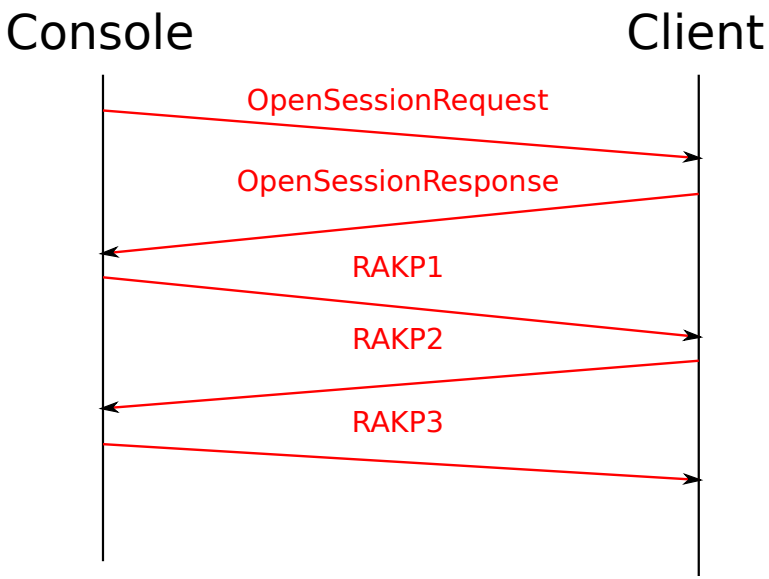
- les messages sont encapsulés dans UDP ;
- le trafic peut être envoyé sur :
  - le port `legacy` (623) : pas d'authentification, pas d'intégrité,
  - le port `secure` (664) : les messages RMCP sont encapsulés dans RSP ;
- la carte intercepte le trafic sur ces ports, analyse les paquets RMCP et répond aux requêtes si nécessaire.

La carte réseau doit donc implémenter toute la pile réseau : IP/UDP/RSP/RMCP.

ANSSI

RMCP Security-Extensions Protocol (RSP)

- RSP ajoute l'authentification mutuelle de la console (le poste d'administration) et du client ;
- les messages sont protégés par l'utilisation de clés symétriques HMAC-SHA1 ;
- ouverture de session et négociation de clé de session via le protocole RSSP (RSP session protocol).



ANSSI

Support du RMCP par les matériels

- HP Compaq dc7600
  - les messages de démarrage, arrêt et redémarrage fonctionnent sur le port *secure*.
- DELL Latitude D530 et Precision T5400
  - le message CapabilitiesRequest est correctement pris en compte,
  - le message CapabilitiesReply indique que les fonctions d'administration à distance ne sont pas supportées.

Remarques

- aucun matériel ne supporte l'administration à distance sur le port *legacy* non sécurisé ;
- pourquoi certains vendeurs ont-ils désactivé les fonctions d'administration à distance tout en gardant une implémentation fonctionnelle du protocole RMCP ?

ANSSI

Comportement de la carte avec ASF

À la réception d'un paquet, la carte :

- intercepte le paquet **avant** transmission au système ;
- vérifie s'il s'agit d'un paquet RMCP ;
- le traite :
  - ouverture et fermeture de session,
  - envoi de l'état du système,
  - administration du système.

Dans ce cas, le paquet n'est **pas** transmis à l'hôte.

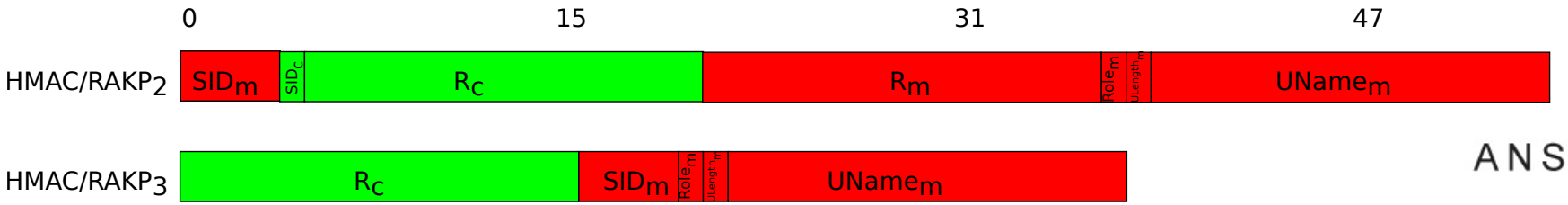
ANSSI

Problèmes potentiels

- le protocole utilise des clés partagées de 160 bits ;
- les messages sont protégés en intégrité mais le calcul n'inclut pas d'identifiant de message ;
- pour se faire passer pour la console, un attaquant a uniquement besoin d'envoyer un $\texttt{RAKP}_3$ avec un `HMAC` valide.
  - peut on forger le `HMAC` ?
  - peut on utiliser un client comme un oracle ?

En pratique

- l'attaquant ne contrôle pas tous les champs des messages ;
- la taille des champs pose problème.



ANSSI

Champs intéressants sous le contrôle d'un attaquant

- *username* de la console ($\texttt{RAKP}_1$) ;
  - les spécifications limitent la taille à 16 caractères, sans `NULL` ;
  - mais la taille du champ est codée sur un octet.
- *session ID* de la console (`Open Session Request`).

Et si l'attaquant ne respecte pas les spécifications ?

Suivant les *firmwares* et le contenu des champs :

- les sessions sont corrompues (`HMAC` invalides) ;
- la carte « plante » : le système peut envoyer des trames *ethernet* mais ne peut plus en recevoir.

Que se passe-t-il exactement ?

ANSSI

Preuve de concept

- Est-ce une vulnérabilité sérieuse ?
- Quelles en sont les conséquences (directes et indirectes) ?
- Comment construire une preuve de concept ?

Changement d'orateur

ANSSI

Instrumentation de la carte

Comment comprendre ce qu'il se passe ?

- qu'est ce qui plante ?
- comment est-ce que cela plante ?

La documentation *NetXtreme*

- des documents publics sont disponibles pour les développeurs de pilotes *open-source* ;
- ils décrivent les comportements internes de la carte ;
- ils donnent des information à propos du processeur `RX RISC`.

ANSSI

Quelles sont les informations disponibles ?

D'après les spécifications et expérimentations :

- un registre de mode ;
- un registre d'état ;
- le *program counter* ;
- un registre de *breakpoint* ;
- des registres généraux.

On peut utiliser ceci pour construire un *debugger* de carte réseau.

ANSSI

Notre *debugger* maison :

- utilise les registres projetés en mémoire centrale ;
- fait du pas à pas ;
- trace les accès aux registres et à la mémoire interne ;
- peut s'interrompre lors d'un accès à un registre ou une case mémoire spécifique ;
- fait de la recherche de motif.

ANSSI

Aide

```
What should I do next (h for help)? h
Usage:
'a' -> Advance n steps                          'T' -> Track register
's' -> Advance 1 step                           'L' -> Track memory address
't' -> Trace                                    'Z' -> Track specific memory zone access
'c' -> Continue                                 'I' -> Track pattern in memory
'C' -> Continue (step-by-step                   'P' -> Track pattern
'g' -> Break on instruction                     'x' -> Clear tracking
'R' -> Break on pattern in register             'f' -> Find pattern in internal memory
'S' -> Break on pattern in stack                'F' -> Find pattern in external memory
'H' -> Break on pattern in internal memory      'A' -> Find all patterns in external memory
'M' -> Break on pattern in external memory      'd' -> Display memory address
'n' -> Break on next pattern in stack           'D' -> Display memory area
'l' -> Break on specific memory access          'w' -> Write a word to memory address
'm' -> Break on any memory access               'r' -> Reset CPU
'j' -> Break on register write                  'q' -> Quit
'i' -> Break on instruction
```

ANSSI

CPU

```
******** Instruction
Instruction = 3c020001 LUI r2 = 00010000
Last memory access = 00000000
******** CPU Status Registers ******
RXPC    = 00011078 RXHWBRK = 0000001d
RXMODE = 00009db0 RXSTATE = 80001400
```

ANSSI

Registres généraux

```
******** CPU Registers ************************************
$0  = 00000000 $1   = 00010000 $2   = 00000000 $3  = 40000000
$4  = 0001b4b8 $5   = 0001b8e6 $6   = 00000000 $7  = 0001bfc4
$8  = 00000040 $9   = 00000050 $10  = 0001b8bc $11 = 0001bfc0
$12  = 80000000 $13 = 00000001 $14  = 00000000 $15 = ffffffbf
$16  = a4020000 $17 = aaaaaaaa $18  = 00000000 $19 = 0001af48
$20  = 0000ad60 $21 = 018004f1 $22  = 000000fc $23 = 00010000
$24  = ffffffff $25 = 80000000 $26  = 00000b50 $27 = 00011104
$28  = c0000000 $29 = 0001bfd8 $30  = 0001c000 $31 = 000111f8
```

ANSSI

Pile

```
******** Stack ******************************
Stack pointer: 0001bfd8 (max) stack size: 10
Stack bottom:  0001c000
************************************************
0001bffc:73ffffff  0001bfe8:00010e00
0001bff8:00010044  0001bfe4:0001a918
0001bff4:0001a000  0001bfe0:0000ad60
0001bff0:0000ad64  0001bfdc:0001a000
0001bfec:0001a80c  0001bfd8:00010b3c
```

ANSSI

Pourquoi la carte plante-t-elle ?

Le registre d'état du `RX RISC` fournit des indications sur la raison d'un crash :

1. `bad memory alignment`
2. `invalid data access` (accès à une adresse mémoire invalide) ;
3. `invalid instruction fetch` (saut à une adresse invalide) ;
4. `invalid instruction` ;

Les points 3 et 4 correspondent à un détournement direct du flot d'exécution.
Les points 1 et 2 peuvent probablement mener à un détournement indirect du flot d'exécution via l'écrasement d'une adresse de retour sur la pile.

ANSSI

Détournement du flot d'exécution

Quand le CPU `RX RISC` s'arrête, l'attaquant doit :

- trouver la source de la donnée fautive ;
- l'ajuster pour arriver à ses fins.

Par expérimentation en aveugle, on finit par :

- trouver un débordement de tampon dans le champ `username` ;
- écraser une adresse de retour dans la pile avec une adresse sous notre contrôle.

ANSSI

Preuve de concept d'injection de code (1/2)

Sur le modèle de carte étudiée, pour le modèle de *firmware* donné, un attaquant peut exécuter du code arbitraire :

Saut initial

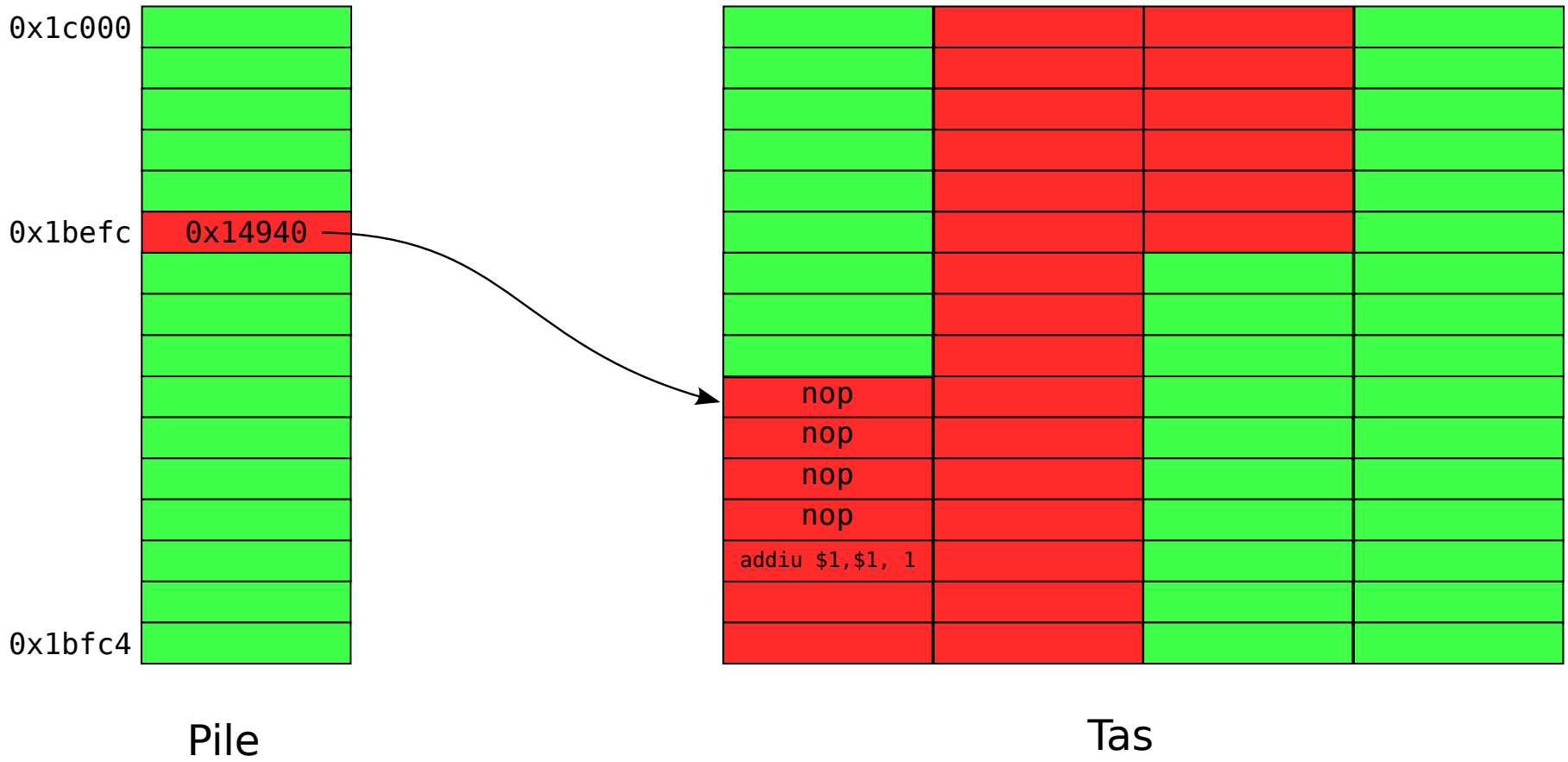
- un attaquant peut écraser une adresse de retour dans la pile ;
- peut trouver l'adresse (stable pour ce *firmware*) de *username* ;
- peut donc mettre du code d'exploitation dans *username* et écraser l'adresse dans la pile par celle-ci.

1er étage

- le champ *username* fait 255 caractères, (peu d'instructions) ;
- mais l'attaquant a accès aux tampons réseau ;
- il peut donc placer du code dans un paquet **avant** l'attaque et y sauter.

ANSSI

Code d'exploitation, premier étage



ANSSI

Preuve de concept d'injection de code (2/2)

Deuxième étage

Le deuxième étage se trouve dans les tampons réseau :

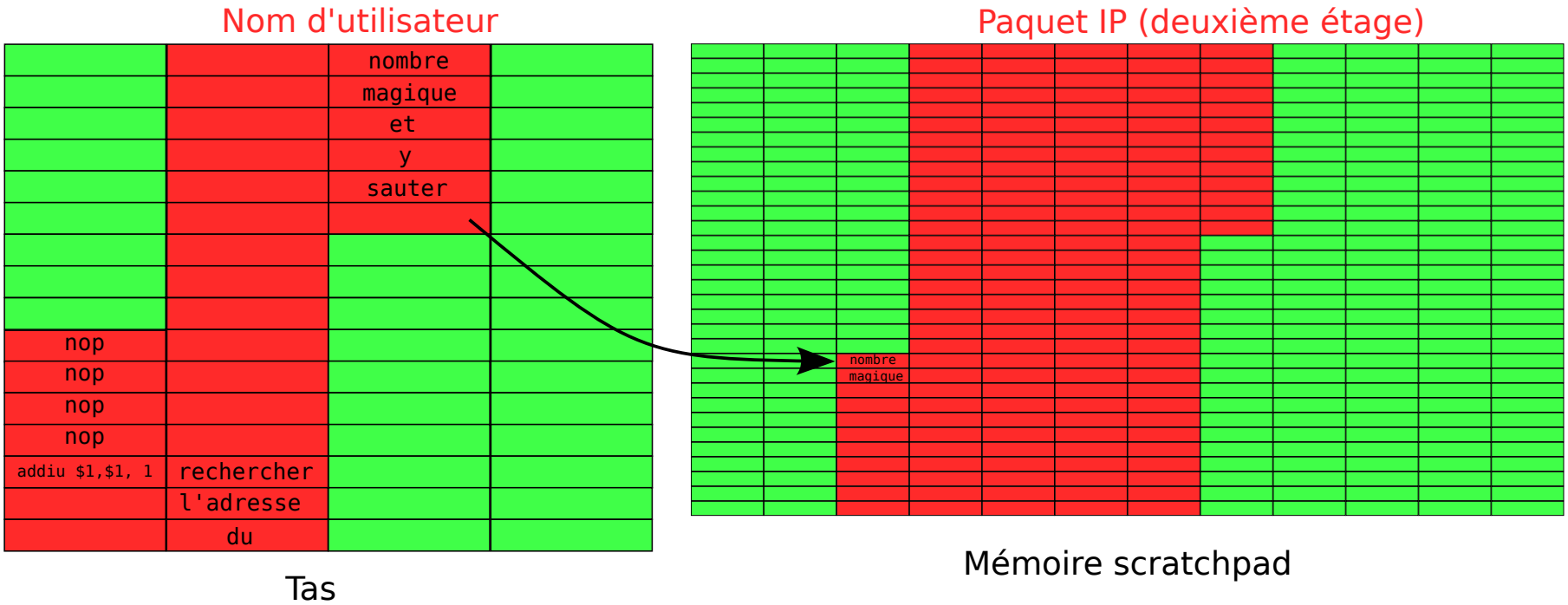
- d'une taille largement suffisante pour un attaquant ;
- envoyé comme un paquet standard avant le code d'exploitation ;
- commence par un motif connu afin que le premier étage puisse le trouver.

L'attaquant peut maintenant :

- exécuter du code arbitraire sur le `RX RISC` ;
- fournir à la volée du code nouveau en utilisant de simples paquets réseau ;
- réécrire le *firmware* si besoin ;
- ...

ANSSI

Stage 2


Nom d'utilisateur
nombre
magique
et
y
sauter
nop
nop
nop
nop
addiu $1,$1, 1
rechercher
l'adresse
du
Tas
Paquet IP (deuxième étage)
nombre
magique
Mémoire scratchpad

Man in the middle

Tous les paquets passent par la mémoire de la carte :

- les paquets reçus juste avant d'atteindre l'hôte ;
- les paquets à émettre juste avant d'être émis sur le cable.

On peut :

- dérouter le trafic `DNS` ;
- dérouter tout le trafic ;
- modifier des négociations `TLS` ;
- réaliser n'importe quel `MITM` de façon silencieuse.

ANSSI

Contrôle à distance

DELL avait désactivé les fonctions de contrôle :

- la carte est toujours connectée au SMBus ;
- la table ACPI ASF est présente, avec les fonctions de contrôle à distance ;
- le *firmware* peut envoyer des messages précis sur le SMBus ;
- le *firmware* peut donc démarrer, arrêter, redémarrer la machine.

L'attaquant peut ainsi rétablir le contrôle à distance sur les machines DELL !

ANSSI

Prise de contrôle de l'hôte

La carte réseau :

- est sur le bus PCI/PCI-Express ;
- peut lire et écrire dans l'espace de configuration PCI ;
- a un accès direct à la mémoire de l'hôte (DMA).

Un attaquant peut donc lire et écrire dans la mémoire centrale.

ANSSI

Via le DMA

Transferts DMA

- la carte et l'hôte s'échangent les paquets via le DMA ;
- les méta-données (adresse sur la carte, adresse dans l'hôte, taille des tampons) sont stockées dans des structures spéciales, les *buffer descriptors*.

preuve de concept : écriture dans la mémoire centrale

- écriture d'une adresse hôte dans un *buffer descriptor* de la carte ;
- envoi de paquets sur le lien ;
- le paquet est écrit dans la mémoire centrale à l'adresse voulue par l'attaquant.

ANSSI

L'attaque dépend du système

- comme toutes les attaques basées sur le DMA ;
- doit contourner une éventuelle protection mémoire (IOMMU/VT-d) ;
- doit trouver où lire/écrire ;
- doit déclencher l'exécution du code.

- utilisation de Linux pour la preuve de concept ;
- l'attaque fonctionnerait de même sur un autre système ;
- astuce : on configure une nouvelle adresse MAC sur la carte : 90:90:90:90:90:90

ANSSI

Démonstration

Ce qui est fait dans la démonstration :

- écriture de code permettant de lancer un `shell` distant à l'adresse 0 en mémoire centrale ;
- détournement de `icmp_rcv` pour passer à l'adresse 0 en cas de réception d'un paquet `ICMP` particulier ;
- envoi d'un ping magique.

ANSSI

Contre-mesures

- utiliser un *firmware* corrigé ;
- désactiver `ASF` (et pas seulement dans le `BIOS`) ;
- filtrer les ports `UDP` `ASF` et `RMCP` ;
- utiliser `IOMMU/VT-d` sur un système compatible ;
- désactiver les protocoles d'administration à distance, les réserver à des réseaux séparés et sûrs.
  - de toute façon personne n'aurait eu l'idée d'activer `ASF` sur un portable connecté à internet.
  - (Espérons le)

ANSSI

Conclusion

La vulnérabilité peut sembler inquiétante, mais :

- peu de cartes supportent ASF ;
- encore moins ont ASF activé.

Cependant,

- ASF est un protocole très simple :
  - basé sur UDP,
  - peu d'algorithmes cryptographiques,
  - sessions en nombre limité,
  - pas d'interaction avec le reste du système d'information ;
- AMT, IPMI, et les autres protocoles d'administration à distance sont nettement plus complexes :
  - au-dessus de TCP,
  - utilisation des *webservices* (XML-RPC, SOAP...),
  - interaction avec le système d'information (*Active Directory*, *Kerberos*...).

ANSSI

Conclusion (2/2)

- de plus en plus de périphériques utilisent des *firmwares* :
  - cartes réseau,
  - cartes sans-fil,
  - puces `GSM` et `UMTS`,
  - contrôleurs `RAID` et `SSD`.
- avec des caractéristiques communes :
  - pas de code source disponible,
  - code très proche du matériel,
  - accès au monde extérieur (cartes réseau),
  - contraintes temps réel,
  - non soumis au cloisonnement et mesures de sécurités liées au système d'exploitation.

De plus en plus de problèmes sont à prévoir dans le futur.
Le développement de cartes et périphériques plus simples est-il possible ?

ANSSI

Questions – réponses

?

Une FAQ est disponible sur

http://www.ssi.gouv.fr/trustnetworkcard

ANSSI